Omega 43 (2014) 96–107

Contents lists available at ScienceDirect

Omega

journal homepage: www.elsevier.com/locate/omega

A combined robot selection and scheduling problem for flow-shops with no-wait restrictions

Dvir Shabtay*, Kfir Arviv, Helman Stern, Yael Edan

Department of Industrial Engineering and Management, Ben-Gurion University of the Negev, Beer-Sheva, Israel

ARTICLE INFO

Article history:
Received 11 July 2012
Accepted 17 July 2013
Available online 30 July 2013

Keywords:
Scheduling
Flow-shop
No-wait restriction
Robotic flow-shop
Makespan
Bicriteria shortest path problem

ABSTRACT

This paper addresses a bicriteria no-wait flow-shop scheduling problem with multiple robots transferring jobs between pairs of consecutive machines. The robots share an identical track positioned alongside the machine transfer line. Each robot is assigned to a portion of the tract from which it performs job transfers between all reachable machines. We assume that job processing times are both machine and job independent, that jobs are not allowed to wait between two consecutive machines and that machine idle times are not allowed. We define a combined robot selection and scheduling problem ($\mathcal{RSSP}$) for a set of Q non-identical robots characterized by different costs and job transfer and empty movement times. A solution to the $\mathcal{RSSP}$ problem is defined by (i) selecting a set of robots, (ii) assigning each robot to a portion of the track, and (iii) scheduling the robot moves. We define a robot schedule as feasible if all the jobs satisfy the no-wait condition and there are no machine idle times. The quality of the solutions are measured by two criteria (performance measures): makespan and robot selection cost. We study four different variations of the $\mathcal{RSSP}$, one which is shown to be solvable in polynomial time while the other three turn out to be $\mathcal{NP}$-hard. For the $\mathcal{NP}$-hard, we show that a pseudo-polynomial time algorithm and a fully polynomial approximation scheme exists, and derive three important special cases which are solvable in polynomial time. The $\mathcal{RSSP}$ has aspects of robot selection, machine-robot assignment and robot movement scheduling. We believe this is the first time that this type of problem has been treated in the literature, and addresses a very important problem in multiple robotic systems operation. Our contribution lies in the formulation, methodology, algorithms for solution and complexity results which jointly treats all aspects of the problem simultaneously without the need to defer to heuristic decomposition methods.

1. Introduction

Robotic flow-shop systems, which consist of a set of robots that are responsible for transferring jobs between pairs of consecutive machines, widely appear in automatic manufacturing systems. Any savings in cost and time of these manufacturing transfer lines enhances the competitiveness of world class companies. The problem addressed in this paper has the objective of reduced manufacturing time and costs. In solving such systems it is common practice to assume that robot selection and assignment decisions are considered to be at a higher hierarchal decision level than the actual robot scheduling decisions. Although higher and lower level decision-making processes are tightly connected, traditional robot scheduling problems have been extensively studied under the assumption that any higher level decision has already been made; and that the set of robots and their assignment to machines are predefined to the scheduler. This decoupling of the higher and lower processes provides solutions whose improvement can save companies substantial savings in time and cost. In this paper we aim to provide an original methodology that efficiently coordinates between these two very important and closely related decisions by solving them simultaneously. Our results should provide manufactures the opportunity to enhance the performance of their productive endeavors.

This paper addresses a bicriteria *no-wait* flow-shop scheduling problem with multiple robots transferring jobs between pairs of consecutive machines. The problem is formally stated as follows: a set of n independent jobs, $J=\{J_1,\ldots,J_n\}$, is available for processing at time zero. The jobs are to be processed in a fixed order on a set of m machines, $M=\{M_1,\ldots,M_m\}$ in a flow-shop scheduling system ($n\geq m-1$). In such a system, each job J_j consists of m operations $O_j=\{O_{1j},\ldots,O_{mj}\}$ which must be processed in the order $O_{1j}\rightarrow O_{2j}\rightarrow\cdots\rightarrow O_{mj}$. The operation O_{ij} must be processed on M_i without preemption for $p_{ij}\geq 0$ time units. It is assumed that the processing times are both job- and machine-independent, i.e., $p_{ij}=p$

* Corresponding author. Tel.: +972 8 6461389.
E-mail addresses: dvirs@bgu.ac.il (D. Shabtay), arvivk@bgu.ac.il (K. Arviv), helman@bgu.ac.il (H. Stern), yael@bgu.ac.il (Y. Edan).

0305-0483/$ - see front matter
http://dx.doi.org/10.1016/j.omega.2013.07.001

($i=1,\ldots,m$; $j=1,\ldots,n$). Each machine can process only one job at a time. There are Q types of robots where the cost of a single robot of type q is δ_q ($q=1,\ldots,Q$). Let t_{ijq} be the transportation time required for a type q robot to transfer J_j from M_i to M_{i+1}. It is assumed that the times are job-independent such that $t_{ijq}=t_{iq}$. Empty return times te_{iq} are also defined when a type q robot moves empty without carrying a job from M_{i+1} to M_i. The empty return times are assumed to be additive, i.e., the time for the robot to travel between two distinct machines is the sum of the empty traveling times between all intermediate machines. For the case of a single robot type we omit the q index such that, for example, t_i is the transportation time required to transfer any job from M_i to M_{i+1}. It is also assumed that there is an automatic mechanism beside each M_i, which allows each robot to perform download/upload operations in a negligible time without violating the no-wait restrictions.

Physically robots are constrained to move on an identical track, positioned alongside a machine transfer line. Due to the limited working space envelopes of the robots and to avoid collisions, each robot is assigned to a portions of the track and performs job transfers between all reachable machines from its assigned portion of the track. Let k be the number of robots serving the system and let $\mathcal{M}_r=\{M_{l_r},\ldots,M_{l_{r+1}}\}$ be a subset of consecutive machines assigned to R_r ($r=1,\ldots,k$) with $l_1\overset{\text{def}}{=}1$ and $l_{k+1}\overset{\text{def}}{=}m$. R_r is responsible for transferring jobs between successive machines in $\mathcal{M}_r$. Note, $\mathcal{M}_r\cap\mathcal{M}_{r+1}=M_{l_{r+1}}$ and R_r is responsible for transfers to $M_{l_{r+1}}$ while R_{r+1} is responsible for transfers from $M_{l_{r+1}}$ for $r=1,\ldots,k-1$. Fig. 1 below illustrates the flow-shop system and its machine partition based on k robot working space envelopes.

The robot selection and scheduling problem (referred to as $\mathcal{RSSP}$ in short) includes two parts. The first is the robot selection and assignment, and the second is the robot scheduling. The robot selection and assignment part is composed of a selection of an ordered list of k robots $\{R_1, R_2,\ldots,R_k\}$ (where k itself is a decision variable) and from the assignment of a subset of machines, $\mathcal{M}_r=\{\mathcal{M}_{lr},\ldots\mathcal{M}_{lr+1}\}$, to each robot R_r for $r=1,\ldots,k$ such that $\mathcal{M}_1\cup\mathcal{M}_2\cup\ldots\cup\mathcal{M}_k=\{M_1,\ldots,M_m\}$. The robots are selected from the set of Q robot types and more than a single robot of the same type may be selected. The scheduling part defines a set of *moves* for each robot which indicates the sequence in which the robot serves the machines. Similar to Che and Chu [12], a solution to the scheduling part is *feasible* if it obeys the following two restrictions:

Restriction 1: (no-wait restriction): jobs are not allowed to wait between two consecutive machines, that is, once a job has finished its processing on M_i it must be immediately transferred to M_{i+1} ($i=1,\ldots,m-1$).

Restriction 2: (no machine idle time): once a machine has started work, it must process the entire set of n jobs consecutively.

The quality of a feasible solution to $\mathcal{RSSP}$ is evaluated by two different performance measures. The first is the makespan criterion denoted by $C_{\max}=C_n$, and defined by the completion time of the last job (J_n) on the last machine (M_m). The second is the total cost of the assigned robots defined by

$$TRC=\sum_{r=1}^{k}\delta_{\{r\}}$$

where $\{r\}$ is the type of robot R_r.

In any multicriteria problem it is important to point out the nature of the optimization being performed, as different criteria

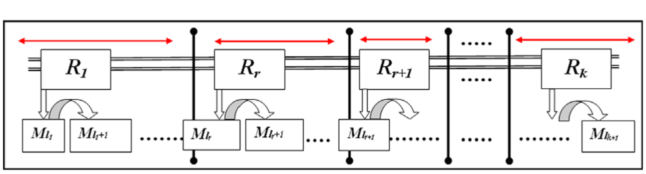

Fig. 1. An illustration of the robot job-machine scheduling system.

are often conflicting. This is reflected by the following four variations of the $\mathcal{RSSP}$:

1. $\mathcal{RSSP}_1$: find a feasible solution which minimizes the total integrated cost, i.e., $C_{\max}+TRC$.
2. $\mathcal{RSSP}_2$: find a feasible solution which minimizes $C_{\max}$ subject to $TRC\le\overline{TRC}$, where $\overline{TRC}$ is a given upper bound on the total robot assignment cost.
3. $\mathcal{RSSP}_3$: find a feasible solution which minimizes TRC subject to $C_{\max}\le\overline{C}_{\max}$, where $\overline{C}_{\max}$ is a given upper bound on the makespan value.
4. $\mathcal{RSSP}_4$: identify a Pareto-optimal solution for each Pareto-optimal point, where a feasible solution S is called *Pareto-optimal* (*non-dominated* or *efficient*) with respect to criteria $C_{\max}$ and TRC; if there does not exist another feasible solution S' such that $C_{\max}(S')\le C_{\max}(S)$ and $TRC(S')\le TRC(S)$, with at least one of these inequalities being strict.

Note that solving problem $\mathcal{RSSP}_4$ also solves problems $\mathcal{RSSP}_1-\mathcal{RSSP}_3$ as a by-product. Note also that the decision version (DV) of problem $\mathcal{RSSP}_2$ is identical to that of problem $\mathcal{RSSP}_3$, and is defined below:

Definition 1. DV: given parameters $\overline{C}_{\max}$ and $\overline{TRC}$, is there a solution for the $\mathcal{RSSP}$ with $C_{\max}(S)\le\overline{C}_{\max}$ and $TRC(S)\le\overline{TRC}$?

The fact that both the $\mathcal{RSSP}_2$ and the $\mathcal{RSSP}_3$ problems share the same decision version implies that either both or none of them is $\mathcal{NP}$-hard.

Robotic flow-shop systems are very complicated and therefore researchers applied various simplified assumptions to provide an analytical analysis. The most common assumption is that there is a single robot that serves the entire production line (see e.g., [52,49,34,30,5,5,22,41]). Among the other commonly used assumptions are: (a) the number of machines is limited to two (see e.g., [54,15,28,5,22]), (b) empty return times equal zero (see e.g., [29,22,41]), (c) loading and unloading times are zero (see e.g., [29,5,6]), (d) job processing times are job-independent (see e.g., [35,25,8,9,14]), (e) the production is cyclic (see e.g., [35,25,5,8,9,14]), and (f) there is sufficient number of robots with no technological constraints (see e.g., [54,15,28]).

A good example where a wide range of assumptions used is the paper by Hurink and Knust [22], where it is assumed that only a single robot serves the entire set of machines and that there is an unlimited buffer between pair of consecutive machines. Moreover, in several cases they even consider more restricted models of two machines, equal transportation time, zero empty return times, and equal and even unit processing times. We note that the robotic flow-shop scheduling problem is so complicated that even under these very restrictive assumptions Hurink and Knust [22] showed that the problem remains strongly $\mathcal{NP}$-hard in most cases. Moreover, they were only able to present polynomial time algorithms for special cases where *all* processing times are *equal*.

The model presented above (in the Introduction section) includes some new features that have not (or rarely) been discussed at all in the literature. Among those features are several robot types, an arbitrary number of machines, a possibility to control the number of robots assigned to the production line and their assignment to machine sets, and a bicriteria objective function. On the other hand, to be able to provide a mathematical based analysis the following assumptions are used: (i) there are no-wait and no-idle restrictions (see Restrictions 1 and 2), (ii) transportation times are job-independent, and (iii) job processing times are *all* equal. The no-wait and no-idle restrictions are well justified by many real-life applications (see e.g., [12,20,51,19]) and are widely used in the flow-shop scheduling literature ([1,24,47,33,35,5,6,8,9,37,38,3,36] and among many others). Even